



SAVONIA

■ OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

REMONTTIMARKKINAT

Välityspalvelu

TEKIJÄ: Henri Harjanne

Koulutusala			
Tekniikan ja liikenteen ala			
Koulutusohjelma			
Tietotekniikan koulutusohjelma			
Työn tekijä(t)			
Henri Harjanne			
Työn nimi			
Remonttimarkkinat - Välityspalvelu			
Päiväys	09.04.2018	Sivumäärä/Liitteet	29
Ohjaaja(t)			
Lehtori Keijo Kuosmanen, Lehtori Jussi Koistinen			
Toimeksiantaja/Yhteistyökumppani(t)			
Entecore Ky			
Tiivistelmä			
<p>Opinnäytetyössä suunniteltiin ja toteutettiin remonteille selainpohjainen välityspalvelu. Välityspalvelun tarkoituksena on helpottaa niin remontin tekijöiden kilpailutusta kuin remontointia tarjoavan yrityksen etsimistä. Työ toteutettiin Entecore Ky:n palveluksessa ja työn tilaajana oli Horizon Interactive Oy.</p> <p>Teoria osuudessa kerrotaan selainpohjaisista tekniikoista ohjelman eri osa-alueilla ja projektihallinnan viitekehyksien heikkoudesta ja hyödyistä. Lisäksi käydään läpi, kuinka projektia voidaan hallita tehokkaasti. Välityspalvelun kehitys toteutettiin asiakkaan kanssa tiiviissä yhteistyössä käyttäen ketterän kehityksen Scrum-mallia auttamaan projektin pysymistä aikataulussa. Lopputuloksena Horizon Interactive Oy sai asiakkaan vaatimukset täyttävän ja helposti eri aloille muokattavan sovelluksen, jonka ympärille on mahdollista rakentaa liiketoiminta.</p>			

Avainsanat
Remonttimarkkinat, välityspalvelu, NoSQL, Groovy

Field of Study			
Technology, Communication and Transport			
Degree Programme			
Degree Programme in Information Technology			
Author(s)			
Henri Harjanne			
Title of Thesis			
Remonttimarkkinat – a Web-Based Matching Service			
Date	9 April 2018	Pages/Appendices	29
Supervisor(s)			
Mr. Keijo Kuosmanen, Senior Lecturer			
Mr. Jussi Koistinen, Senior Lecturer			
Client Organisation /Partners			
Entecore Ky			
Abstract			
<p>The purpose of this thesis was to plan and develop a web-based matching service for customers looking for renovation companies and companies looking for renovations. The main idea for the matching service was to help consumers publicly ask for bids and help renovators get clients. The service was commissioned by a company called Horizon Interactive Oy, but it was created while working for Entecore Ky.</p> <p>For the theory part information about web-based solutions on different aspects of the program as well as different types of project management solutions and their strenghts and weaknesses were studied. The service itself was created in tight co-operation with the commissioner using SCRUM to guide the project to stay in the timeline. The service was built carefully with the chosen techniques to ensure a long lifespan and scalability in future. The main techniques to build the service were Java and MongoDB combined together to build the back-end and Javascript was responsible for handling user inputs on the front-end.</p> <p>As a result of this thesis, a working system with all the required functionalities was delivered for Horizon Interactive Oy to build a business around it. The software itself is in a state where it is easily modified for different services.</p>			

Keywords
Remonttimarkkinat, NoSQL, MongoDB, Groovy

ESIPUHE

Kiitos Mikko Antikaiselle (Entecore Ky) luottamuksesta ja mahdollisuudesta työskennellä asiakkaan kanssa itsenäisesti Entecoren edustajana. Kiitos Keijo Kuosmaselle ohjauksesta ja luonnontieteiden kandidaatti Henna Lauroselle kannustamisesta ja avusta.

Henri Harjanne

SISÄLLYSLUETTELO

ESIPUHE.....	6
TERMIT JA LYHENTEET	8
1 JOHDANTO.....	9
2 TEORIA	10
2.1 Välityspalvelut.....	10
2.2 NoSQL-tietokannat	11
2.3 Web-ohjelmoinnin tekniikat	12
2.3.1 Back-end.....	12
2.3.2 Front-end.....	14
2.4 Projektinhallinnan viitekehys.....	14
3 KÄYTETYT TEKNIIKAT	16
3.1 Tietokanta.....	16
3.2 Back-end	16
3.3 Front-end.....	17
3.4 Käyttöliittymän suunnittelu	17
4 REMONTTIMARKKINAT	19
4.1 Tarjouspyynnön luominen.....	20
4.1.1 Tiedostojen lähettäminen.....	21
4.1.2 Käyttäjän rekisteröityminen	22
4.2 Omat sivut	23
4.3 Viestikeskus ja tarjouksen jättäminen.....	23
4.4 Laskutus ja raportointi.....	25
4.5 Kiinnostavat tarjouspyynnöt	26
5 YHTEENVETO	27
LÄHTEET JA TUOTETUT AINEISTOT	28

TERMIT JA LYHENTEET

CSS

Cascading Style Sheets on kieli, jonka avulla määritetään, miten HTML-elementit näytetään näytöllä tai muissa medioissa.

HTML

Hyper Text Markup Language on kieli, joka kuvaa sivujen rakenteen selaimelle.

JavaScript

Komentosarjakieli, jota käytetään pääsääntöisesti dynaamisen toiminnallisuuden lisäämiseksi.

Front- ja back-end

Ohjelman eri tasot: front-end eli ohjelman esitystaso ja back-end eli ohjelman tiedonkäsitteilytaso.

CMS

Sisällönhallintajärjestelmä eli järjestelmä, joka on keskittynyt palvelemaan organisaation tarpeita muillakin kuin vain verkkosivujen osalta.

1 JOHDANTO

Työn tavoitteena oli luoda remonteille välityspalvelu Horizon Interactive Oy:lle, minkä avulla remonttikohteen ilmoitus voidaan yhdistää alueella olevalle urakoitsijalle. Palvelu helpottaa remontoijia löytämään remontille luotettavan tekijän kilpailukykyiseen hintaan ja takaa urakoitsijan luotettavuuden läpinäkyvillä urakoitsijoiden arvosteluilla. Urakoitsijoita palvelu auttaa asiakkaiden löytämisessä.

Perinteisesti kuluttajan on täytynyt itse ottaa yhteyttä urakoitsijaan, jolta hän pyytää tarjouksen. Yhden urakoitsijan tarjouksen perusteella kuluttaja ei pysty vertailemaan, onko mahdollisesti saatu tarjous kilpailukykyinen tai onko urakoitsijayritys luotettava. Halutessaan varmuutta tarjouksen edullisuudesta ja urakoitsijan luotettavuudesta kuluttajan on täytynyt etsiä lisätietoja urakoitsijasta sekä mahdollisesti pyytää toiseltakin urakoitsijalta tarjousta. Remontteihin keskittynyt välityspalvelu kääntää tilanteen ympäri mahdollistaen urakoitsijoiden välisen kilpailun kuluttajan tekemästä tarjouspyynnöstä. Palvelu tuo urakoitsijalle asiakkaita ja asiakkaalle kilpailukykyisen tarjouksen. Yleinen mahdollisuus urakoitsijoiden kommentointiin ja arvosteluun kertoo tilaajalle, onko kyseessä luotettava urakoitsija säästäen aikaa ja rahaa.

Urakoitsijoita varten on suunniteltu innovatiivinen algoritmi, joka yhdistää tarjouksen ja urakoitsijan perustuen maantieteelliseen sijaintiin. Urakoitsijoille algoritmi säästää aikaa, koska palvelu ilmoittaa vain heitä kiinnostavista kohteista, joten ylimääräisiä ilmoituksia kohteista ei palvelun kautta tule. Remontin tilaajalle algoritmi pyrkii löytyämään urakoitsijat läheltä varmistaen kilpailukykyiset tarjoukset oikealla osaamisella varustetuilta urakoitsijoilta. Sovellus on ohjelmointivaiheessa toteutettu ajatellen sovelluksen mahdollista jatkokäyttöä eri aloille.

Selainpohjainen ratkaisu mahdollistaa palvelun käytön päätelaitteesta riippumatta. Selainpohjainen ratkaisu ei myöskään vie palvelusta toimintoja pois, vaan pikemminkin mahdollistaa muun muassa helpon ylläpitämisen ja hakukonepalveluiden käyttämisen palvelun kohdennettuun mainontaan. Tietokoneelle asennettava ohjelmisto heikentää käyttäjäkokemusta verrattuna selainpohjaiseen palveluun, koska palvelun vaatima lataus nostaa käyttäjän kynystä käyttää palvelua.

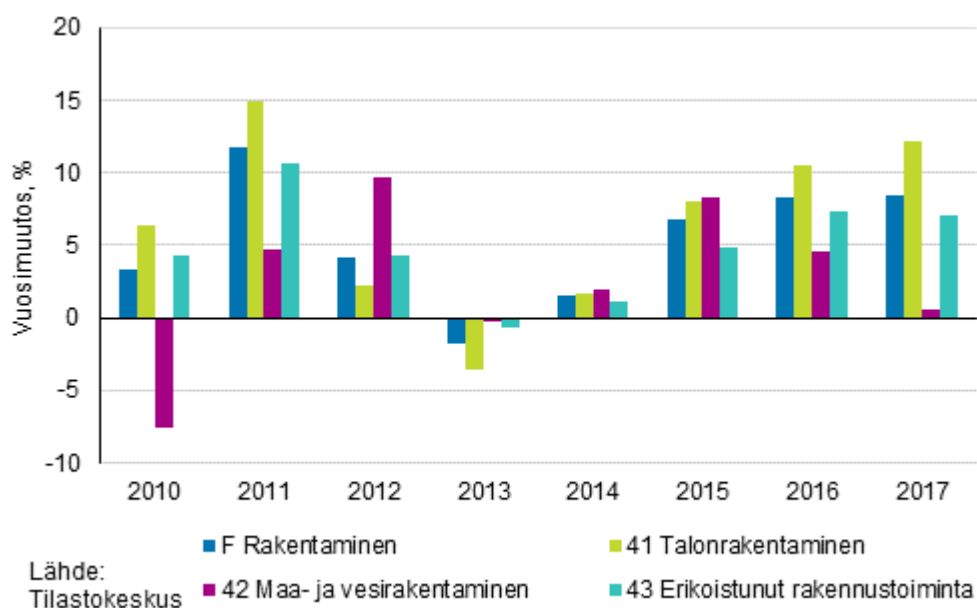
2 TEORIA

2.1 Välityspalvelut

Tuotteiden hintojen vertailu yritysten välillä on kuluttajalle helppoa nykypäivänä varsinkin internetissä ja esimerkiksi eri verkkokaupoista saman tuotteen hinnan hakeminen vaatii vain muutaman hiiren painalluksen. Saatavilla on myös automaattisesti tuotteiden hintoja vertaavia hintojen vertauspalveluja, jotka näyttävät saman tuotteen hinnan eri verkkokaupoissa yhdellä sivulla. Palveluille, kuten remontointi, on hankalaa löytää hinta vain etsimällä tietoa internetistä, sillä palvelun hinta vaihtelee kohteen ja remonttityypin mukaan. Remontin teki- jällä täytyy olla tarpeeksi taustatietoja, mitä tehdään ja millä aikataululla voidakseen muo- dostaa tarjouksen.

Kuluttajalle ja rakennusyritykselle on helpompaa, jos palvelusta saa kohdistetusti kaiken tarvittavan onnistuneen urakkasopimuksen solmimiseen. Remonttipalveluille on olemassa erilaisia palveluita, joissa pystyy kilpailuttamaan remontteja kuten esimerkiksi Alma Media- partners Oy:n <https://www.urakkamaailma.fi/>, joka tarjoaa kuluttajille maksuttoman palve- lun kilpailuttaa oma remontti yli 6 tuhannella urakoitsijalla (Urakkamaailma, 2018). Urak- kamaailma.fi lupaa vähintään neljä tarjousta jokaiseen asiakkaan jättämään tarjouspyyntöön ja luotetut ammattilaiset, joiden tiedot ovat kunnossa. Palvelussa jokainen käyttäjä voi ar- vostella urakoitsijoita muiden näkyville. Kilpailutettujen remonttien arvo on ollut vuoden 2017 aikana yhteensä yli 79 miljoonaa euroa (Urakkamaailma, 2018). Tulot sivustolle tule- vat yhteistyökumppaneille ja mainostajille näkyvyyttä sivuilla ja sosiaalisessa mediassa myymällä sekä urakoitsijoilta hyväksytyistä remonteista veloitettavasta palkkiosta.

Tilastokeskuksen mukaan rakennusyritysten liikevaihto on kasvanut tasaisesti vuodesta 2013 alkaen kokonaisuudessaan reilun 5 prosentin vuosivauhtia (Kuva 1). Remonttien mää- rän kasvua tukemaan tarvitaan kilpailutukseen palvelua, joka helpottaa kuluttajan ja remon- tin tarjoajien yhteistyötä. Viimeisten kuluneiden vuosien perusteella on perusteltu olettaa, että remonttien määrä tulee myös tulevaisuudessa kasvamaan.



Kuva 1: Liikevaihdon vuosimuutosprosentit rakentamisen toimialoilla (Tilastokeskus, 2018)

2.2 NoSQL-tietokannat

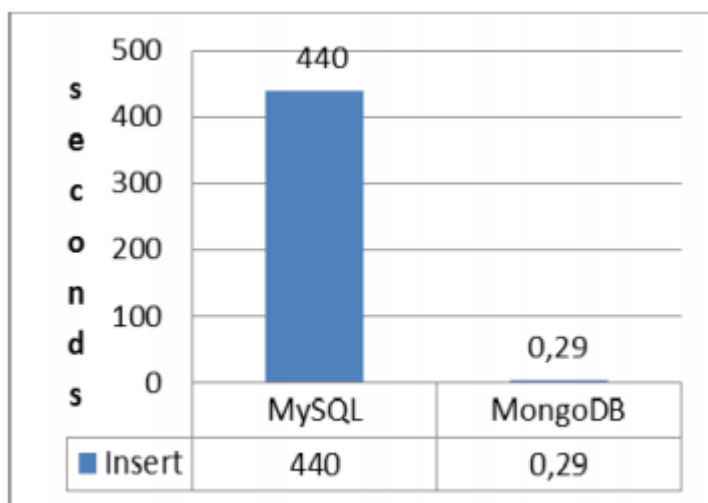
Relaatiotietokannat ovat nykyisin varsin laajasti käytettyjä ja niiden suorituskyky on hyvä rajoitettua datamäärää käsiteltäessä. Valtavia määriä dataa käsitellessä perinteiset relaatiotietokannat ovat tehottomia sekä relaatiotietokannat pystyvät käsittelemään vain rakenteellista dataa, joka on järjestäytynyt riveiksi ja sarakkeiksi. Näihin ongelmiin ratkaisuksi on kehitetty NoSQL-teknologia eli epärelaatiotietokannat (non relational databases). NoSQL-tietokannat pystyvät käsittelemään rakenteetonta dataa, kuten tiedostoja, multimediadataa ja sosiaalisen median dataa, toisin kuin relaatiotietokannat. NoSQL-kannat ovat sekä suorituskyvyltään että tilankäytöltään tehokkaampia kuin relaatiokannat. (Györödi et al., 2015)

NoSQL-teknologioissa on hyödynnetty relaatiotietokantojen ominaisuuksia, jotka ovat tehneet relaatiokannoista suosittuja, ja yhdistetty niihin modernien sovelluksien vaatimia ominaisuuksia. Säilytettyjä asioita ovat: voimakas kyselykieli, konsistenssi ja saumaton integraatio IT-järjestelmiin. (MongoDB)

MongoDB on julkaistu vuonna 2009 ja on tällä hetkellä yksi suosituimmista NoSQL-tietokannoista. MongoDB on joustava tietokannan sisältämän datan tyypin suhteen, sillä se perustuu BSON-dokumentteihin ja dynaamisiin skeemoihin. Tietokannassa data on itsenäistä eli tietokannan skeemat eivät ole relaatiolla yhteydessä toisiinsa. Tämä mahdollistaa dynaamisen skeeman luomisen tietokantaan. Perusoperaatiot, kuten tiedon lisääminen,

päivittäminen ja poisto, eivät ole riippuvaisia rakenteesta ja näin tietokantaa on mahdollista kehittää ketterämmin.

MongoDB:ssä poistuu tarve relaatiotietokannoissa tavanomaisille välitauluille, joihin haetaan tieto toisesta taulusta relaatiolla. MongoDB sisältää mahdollisuuden lisätä skeeman sisälle suoraan toisen skeeman, mikä yksinkertaistaa tietokannan rakennetta. Esimerkiksi tarjouspyyntöä luodessa ei olla riippuvaisia tarjouspyyntö - kuntaid - kunta -rakenteesta, vaan MongoDB:n dynaamisuuden ansiosta tarjouspyyntöön voidaan suoraan sisällyttää kuntatieto ilman skeeman muokkausta.



Kuva 2: Suoritus aika, kun tietokantaan lisättiin 1000 käyttäjää (Györödi et al., 2015)

Györödi et al. (2015) ovat tutkimuksessaan vertailleet MySQL- ja MongoDB-tietokantojen suorituskyykyjen eroa neljällä perusoperaatiolla: lisäys-, kysely-, päivitys- ja poisto-operaatiolla. Jokaisessa suorituskyykytestissä MongoDB:n suoritus aika oli kilpailijaansa nopeampi ja suoritus aikojen ero kasvoi suuremmaksi, mitä isompia tietomääriä operaatioissa käsiteltiin. Esimerkiksi tuhannen käyttäjän tietojen lisääminen MySQL-kantaan kesti 440 sekuntia ja MongoDB-kantaan 0,29 sekuntia (Kuva 2).

2.3 Web-ohjelmoinnin tekniikat

2.3.1 Back-end

Java on vakiinnuttanut asemansa 22 vuoden aikana ja kerännyt taakseen suuren joukon tukijoita ja osaajia. Java on siksi varma ja toimiva valinta sovelluksen pohjalle. Suuri käyttäjäkunta varmistaa nopean virheenkorjauksen ohjelmointikielessä ja suuri joukko tukijoita varmistaa nopean ongelmanratkaisun ohjeiden suuren määrän ansiosta. Java on tällä hetkellä yleisin ohjelmointikieli koko maailmassa (TIOBE).

Groovy on dynaaminen ohjelmointikieli, joka muunnetaan Javaksi ajon aikana. Groovy pohjautuu suurelta osin Javaan, minkä ansiosta Groovy on yhteensopiva kaikkien Java-kirjastojen ja -luokkien kanssa. Kielenä Groovysta on tehty helposti opittava ja dynaaminen. Sen etuja ovat monipuolinen ekosysteemi, tehokkaat ominaisuudet ja tehokas syntaksi. (Groovy)

Groovyssä on erilaisia web-ohjelmointia helpottamaan tarkoitettuja toimintoja. Esimerkiksi kolmen heittomerkin merkkijono muuntaa merkkijonoksi kaiken annetun datan. Näin HTML-koodi pysyy siistinä ja oikein sisennettynä, vaikka se tulostettaisiin Groovyllä. Myös muuttujien lisääminen kyseiseen merkkijonoon on mahdollista sen oman syntaksin avulla eikä tulostettavaa merkkijonoa tarvitse rikkoa osiin tätä varten. (

```
output += """
<div class="col-md-4">
  <a href="/resources/uploads/offeredrelated/${it}" class="lightbox" data-plugin-options="{\"type\":\"iframe\"}">
    
  </a>
</div>
"""
```

Kuva 3)

```
output += """
<div class="col-md-4">
  <a href="/resources/uploads/offeredrelated/${it}" class="lightbox" data-plugin-options="{\"type\":\"iframe\"}">
    
  </a>
</div>
"""
```

Kuva 3: Groovyn sisältämiä toimintoja merkkijonoa luodessa

Play-viitekehys on Java-pohjainen viitekehys, joka pohjautuu Scala-ohjelmointikieleen. Play on yksi käytetyimmistä Java-viitekehyksistä nopeutensa ja massiivisen ekosysteeminsä takia. Play on asynkroninen ja estämätön (non blocking) eli kaikki pyynnöt käsitellään, mutta pyyntöjen ei tarvitse olla järjestyksessä. Esimerkiksi tiedoston generoinnissa kestää kauemmin kuin sivuston lataamisessa, joten tiedoston luomisen aikana voidaan palvella sivun latauksia ja muita nopeita pyyntöjä. Mitään ei myöskään tehdä ilman tarvittavia resursseja, vaan odotetaan resursseja ja tehdään yksi pyyntö kerrallaan. Edellä mainitut asiat mahdollistavat kilpailukykyisen nopeuden ja suuren tukijoukon Play-viitekehysten taustalle myös tänä päivänä. (Play)

Play-viitekehys ja Groovy muodostavat yhdessä ketterän ja tehokkaan ohjelmointiympäristön erilaisille projekteille pohjaksi. Koska Groovyä voidaan ajaa Java-ohjelmien lailla ja Play-viitekehys on täysin Java-pohjainen, voidaan Groovyyn jakaa kaikki tarvittavat luokat ja kirjastot suoraan Play-viitekehyksestä, mikä tekee ohjelmoinnista nopeampaa ja entistä ketterämpää.

2.3.2 Front-end

Bootstrap on CSS-viitekehys, jota lähes kaikki nykyaikaiset selaimet tukevat. Bootstrap helpottaa responsiivisen palvelun luomista kolumnirakenteen avulla. Bootstrapin kolumnit perustavat 12 osaan jaettuun alueeseen, joka muodostaa koko sivun levyisen tilan. Nämä 12 osaa voidaan jakaa ohjelmoijan haluamalla tavalla esimerkiksi kolmeen yhtä leveään osaan eli palstoihin. Tehdyt palstat skaalautuvat automaattisesti erikokoisille laitteille, minkä ansiosta responsiivisen käyttöliittymän tekeminen on yksinkertaista. (Bootstrap) Bootstrap on yksi markkinoiden parhaiten dokumentoituja CSS-viitekehysistä, minkä takia sen suosio on suurta.

JQueryä käytetään käyttöliittymissä luomaan dynaamisuutta ja helpottamaan sivuston käyttöä. Noin 20 prosenttia internetissä olevista sivustoista käyttää jQueryä jollakin tavalla (Build With). JQuery mahdollistaa datan hakemisen sivuston latauksen jälkeen esimerkiksi käyttäjän pyynnöstä ilman, että palvelua täytyy ladata uudestaan. JQuerylle on muutamia kilpailijoita, mutta ne ovat raskaampia ja tarjoavat kokonaisen viitekehysten. Tarpeen ollessa pientä JQueryä voidaan käyttää kevyesti jo rakennetulle sivustolle, jolloin suurta viitekehystä ei ole järkevää ottaa käyttöön. (jQuery)

2.4 Projektinhallinnan viitekehys

Ohjelmistokehityksen kaksi yleisintä projektinhallinnan mallia ovat vesiputousmalli ja ketterät kehitysmallit. Vesiputousmallissa kehitys etenee suoraviivaisesti vaiheesta toiseen seuraavassa järjestyksessä: järjestelmävaatimukset, ohjelmistovaatimukset, analyysi, suunnittelu, ohjelmointi, testaus ja käyttöönotto. Edelliseen vaiheeseen ei mallissa palata missään vaiheessa toisin kuin ketterissä menetelmissä, joissa kehitys etenee iteratiivisesti. Vesiputousmalli edellyttää kattavaa dokumentaatiota ja malli ei ota huomioon vaatimuksien keräämisen jakeen tulleita muutoksia, sillä jokainen kehitysvaihe suoritetaan vain kerran. (Pham & Pham, 2012)

Ketterät menetelmät perustuvat Agile Manifeston periaatteisiin, jotka ovat (Beck et al., 2001):

- Yksilöt ja interaktiot prosessien ja työkalujen edelle
- Toimiva ohjelmisto kattavan dokumentoinnin edelle

- Yhteistyö asiakkaan kanssa sopimusneuvottelujen edelle
- Muutokseen vastaaminen suunnitelman noudattamisen edelle

Ketterä kehitys etenee sprinteissä, joissa jokainen kehitysvaihe toistetaan uudelleen ja näin päästään jokaisessa sprintissä lähemmäs lopputulosta. Ketterän kehityksen etuina voidaan pitää muun muassa seuraavia: riski, että lopputuote ei vastaa asiakkaan vaatimuksiin on pieni; kehitys on kilpailukykyistä; muutokseen sopeudutaan ja tiimit ovat itseohjautuvia. (Pham & Pham, 2012)

3 KÄYTETYT TEKNIIKAT

Remonttimarkkinat on rakennettu käyttäen aikaa kestäviä ja pysyviä ratkaisuja. palvelun pohjaksi on valittu Java-pohjaisia ratkaisuja ja käyttöliittymä on suunniteltu tarjoamaan useaksi vuodeksi toimivan ratkaisun ilman tarvetta päivitykseen.

Sovelluksen kehitykseen tekniikoita valitessa on tärkeää valita tekniikat, jotka ovat kestäviä ja joiden takana on suuri kommuuni varmistamassa pitkän elinkaaren ilman suuria muutoksia. Liian uusi tekniikka voi muuttua jo projektin kehityksen aikana radikaalisti tai pahimmassa tapauksessa sen tuki lopetetaan kokonaan. Liian vanhassa tekniikassa taas voi osa nykyaikaisista toiminnoista olla mahdottomia toteuttaa kokonaan ja jatkokehittäjän löytäminen projektille voi olla hankalaa.

3.1 Tietokanta

Työssä käytettiin tietokantana NoSQL-tekнологiaan perustuvaa MongoDB:tä. Tietokanta valittiin sekä yrityksen käyttämän tietokantapalvelimen, että aikaisemman MongoDB:n käytön hyötyjen ja käyttökokemuksen takia. Työn kannalta suurimpana hyötynä voidaan pitää MongoDB:n sisältämää paikkatietohakua (geospatial search) ja perusoperaatioiden nopeutta. palvelun jatkoa ajatellen MongoDB mahdollistaa skaalautuvuuden sekä rakenteettoman ja rakenteellisen datan tukemisen.

Vaikka MongoDB mahdollistaa hyvän tuen tiedostojen tallentamiseen tietokantaan, tallennettiin palvelun generoimat tiedostot kuitenkin normaaleina tiedostoina palvelimelle. palvelussa olevien sähköpostipohjien ja pdf-tiedostojen laittaminen tietokantaan olisi ollut hankalaa palvelinrakenteen takia, joten päätin luoda yhden selkeän tiedostorakenteen palvelimelle, jossa kaikki palvelun generoimat tiedostot sijaitsevat.

3.2 Back-end

palvelun back-end toteutettiin Bonsait CMS-järjestelmällä, jotta asiakas pystyy tarvittaessa muokkaamaan sivustonsa staattista sisältöä helposti graafisen käyttöliittymän avulla. Bonsait on Play-viitekehystä käyttävä ohjelmisto. Play-viitekehys tarjoaa erilaisia toimintoja helpottamaan ohjelmointia. Play-viitekehysten Java-pohjaisuus mahdollistaa Groovyn yhdistämisen viitekehukseen. Groovy käännettiin sivun renderöintivaiheessa ja se mahdollisti kaikkien Javan mahdollistamien luokkien ja pakettien käytön. palvelun käyttämä logiikka

on toteutettu käyttäen Groovyyä, joka on integroitu Bonsait CMS-palvelun käyttämään Play-viitekehykseen.

3.3 Front-end

Remonttimarkkinoiden palvelusta haluttiin tehdä responsiivinen, jotta sen käyttö olisi mahdollisimman sujuvaa kaiken kokoisilla laitteilla ja selaimilla. Erilaiset CSS-viitekehukset helpottavat responsiivisuuden luomista kuitenkin turhaan rajoittamatta ohjelmointia. Toteutustekniikaksi valikoitu Bootstrap CSS-viitekehys, koska aikaisempi kokemus ja osaaminen viitekehyksestä oli todella vahva eikä siinä ole isoja puutteita, jotka saattaisivat hidastaa lopullisen tuotteen valmistumista.

JQueryä käytettiin palvelun osittain dynaamisen käyttöliittymän ja sen toiminnallisuuksien luomiseen. Osa palvelun toiminnoista ladataan reaaliaikaisesti sivustolle ilman sivuston uudelleen lataamisen tarvetta. Tämä nostaa sivuston käyttäjäkokemusta eikä käyttäjän tarvitse odottaa sivuston uudelleen lataamista turhaan.

3.4 Käyttöliittymän suunnittelu

Remonttimarkkinoiden käyttöliittymän suunnittelimme yhteistyössä Mediatoimisto Koneen kanssa Mediatoimisto Koneen suunnitteleman graafisen ilmeen pohjalta vastamaan asiakkaan vaatimuksia. Käyttöliittymä tukee palvelun responsiivisuutta sekä ohjaa käyttäjää käyttämään palvelua oikein. Lisäksi tutkimme suosituimpien internetpalveluiden käyttöliittymiä saadaksemme varmuutta, millaiseen asetteluun kuluttajat ovat tottuneet tehdäksemme palvelusta mahdollisimman intuitiivisen käyttää.

Käyttöliittymäsuunnittelun kulmakivinä käytimme Ewerin (2016) määrittelemiä sääntöjä käyttöliittymän tarpeille. Määritellyistä säännöistä palveluun valittiin viisi tärkeintä:

1. Ymmärrä käyttöliittymän merkitys käyttäjäkokemukselle – Käyttöliittymästä haluttiin luoda visuaalisesti hieno ja nykyaikainen. Hyväkin käyttöliittymä voi kärsiä huonosti suunnitellusta ulkoasusta.
2. Tiedosta kohderyhmä – Remonttimarkkinat-palvelu on suunniteltu urakoitsijoille ja remonttia tarvitseville ihmisille eli tavoitteena oli rakentaa helppo ja toimiva käyttöliittymä. Urakoitsijoille tärkeää oli myös turhien ilmoitusten eliminointi, minkä takia palvelu lähettää ilmoituksia vain kohdennetuista tarjouspyynnöistä.
3. Pidä asiat yksinkertaisina ja yhdenmukaisina – Yksinkertaisuus ja yhdenmukaisuus helpottavat käyttäjää hyväksymään uuden käyttöliittymän. Palvelussa muun muassa

painikkeet ovat hyvin samankaltaisia, mikä auttaa käyttäjää ymmärtämään, koska häneltä vaaditaan painallusta jatkaakseen palvelussa eteenpäin.

4. Implementoi visuaalinen hierarkia – Tärkeimmät asiat palvelussa pitää tuoda visuaalisesti selkeästi esille. Otsikot ja tärkeimmät painikkeet ovat selkeästi korostettuina verrattuna vähemmän tärkeisiin painikkeisiin.
5. Yksinkertaista lomakkeet – Turhien lomakkeiden tai nappien lisäämistä sivustolle tulisi välttää.

Näiden periaatteiden mukaan suunniteltiin esimerkiksi tarjouspyynnön luomiseen sivu.

Suunnittelussa pyrittiin luomaan samanarvoiset painikkeet samanlaisiksi. Mahdollisimman selkeät ja johdonmukaiset värit pitävät ohjelman yksinkertaisena.

4 REMONTTIMARKKINAT

Palvelun tarkoitus on helpottaa yksityisten remonttien kilpailutusta ja urakoitsijoiden asiakkuuksien solmimista. Palvelussa on mahdollista tarjouspyyntöjen listaus ja urakoitsijoiden arvostelu tehdyn työn perusteella. Mitä tarkemmat tiedot tarjouspyyntöön ilmoitetaan, sitä varmemmin saadaan kohdistettua työ juuri oikealle tekijälle. Urakoitsijoille on myös luotu mahdollisuus kysyä lisätietoja tarjouspyyntöön liityen.

Mediatoimisto Koneen kanssa yhteistyössä luotu käyttäliittymä ja grafiikat tuovat palveluun helposti lähestyttävän ilmeen, värit ja selkeän kuvauksen palvelusta (Kuva 4). Haastellisinta käyttäliittymäsuunnittelussa oli luoda mahdollisimman yksinkertainen käytettävyys kuitenkin säilyttäen tarpeeksi tietueita datan keräämistä varten.

Kehitys mukaili ketteriin kehitysmenetelmiin kuuluvaa Scrum-mallia. Kehityssprinttien päätteeksi pidettiin asiakkaan kanssa palavereita, joissa arvioitiin kehityksen suuntaa ja kehitettyjä osia. Palavereiden päätteeksi suunniteltiin, mitä tehdään seuraavassa sprintissä. Projektia aloitettaessa ei tehty tarkkaa vaatimusmäärittelyä sivustolle, minkä takia projekti osittain kasvoi isommaksi kuin olisi ollut tarpeen.



Kuva 4: Palvelun yleinen ilme ja palvelukuvaus

4.1 Tarjouspyynnön luominen

Käyttäjä päättää ensimmäisten sivustolla vietettyjen hetkien perusteella, onko palvelua miellyttävä käyttää, joten siksi näenkin tarjouspyynnön luomisen olevan tärkein osa palvelua. Koko suunnitteluprosessin ajan pidin mielessä ylimääräisten painikkeiden ja lomakkeiden minimoimisen ilman, että palvelun käytettävyys kärsii. Palveluun rekisteröitymisestä luovuttiin kokonaan ja se yhdistettiin tarjouspyynnön luomiseen, mikä alentaa käyttäjän kynnystä käyttää palvelua. Lisäksi käyttäjiä ilman tarjouspyyntöä ei palveluun otettu mukaan ollenkaan.

Tarjouspyynnön luomista käyttäjälle helpotettiin tekemällä yksinkertaisia ja helppoja valintoja ja painikkeita. Painikkeet suunniteltiin indikoimaan käyttäjälle selkeästi valinnasta tai valinnan puutteesta. Eri valinnat on selkeästi ryhmitelty omiksi alueiksi, millä on saatu selkeät vaiheet tarjouspyynnön luomiseen. Tiivistämällä kenttiä saatiin tarjouspyyntö mahtumaan kokonaan yhteen näkymään eli valintoja ei ole liikaa käyttäjälle (Kuva 5).

Kuva 5: Ilmoita remonttisi -ikkuna, jossa valittuja ja valitsemattomia painikkeita.

Tarjouspyynnön lähettämisen osalta otettiin huomioon ohjelman jatkokehitys ja mahdollinen jatkokäyttö. Tarjouspyynnön luominen on tehty dynaamiseksi, joten kenttiä ja

painikkeita voi asiakas lisätä tai muokata halutessaan helposti. Dynaamisuuden takia tarjouspyynnön lähettämiseen luotiin koodi, joka on mahdollisimman ketterä tulevien muutosten varalta ja joka pystyy generoimaan painikkeille tarpeelliset toiminnot niiden määrästä riippumatta. Ratkaisin asian luomalla omat funktiot, jotka vastaanottavat selektorin parametrinä (Kuva 6).

```

}
$(document).ready(function(){
    renderHandlers(".selectable");
    expand_handler(".expand_button");
    initCounter();
    finishArray(".renderArray");
    getemail(".emailForm");
    goback();
    removefileinputs(".remove_file");
    $('input[type=file]').on('change', prepareUpload);
});

```

Kuva 6: Esimerkki JQuery-funktioiden kutsumisesta

4.1.1 Tiedostojen lähettäminen

Palvelun käyttäjillä on mahdollisuus lähettää toisilleen kuvia muun muassa keskusteluissa ja tarjouspyynnön luomisen yhteydessä. Tiedostoja odotetaan keskimäärin noin kahdesta kolmeen kappaletta per jätetty ilmoitus, minkä takia palvelun logiikan täytyy hidastaa käyttäjän toimintaa mahdollisimman vähän. Parhaaksi ratkaisuksi valitsin tiedostojen lähettämisen taustalla heti, kun ne lisätään lähetettäväksi (Kuva 7). Näin käyttäjän ei tarvitse odottaa tiedostojen lähetystä samaan aikaan, kun lähetetään koko tarjouspyyntöä eteenpäin, vaan tiedostot ovat jo valmiina odottamassa. Ylimääräiset tiedostot poistetaan palvelimelta päivittäin, joten turhat tiedostot eivät aiheuta ongelmia.

```

function sendfiles(){
  var now = new Date();
  var total = $('input[type=file]').length;
  var fileAmountCounter = 0;
  var fileinputcontainsfilecounter = 0;
  var loopcounterforallfileinputswithfile = 0;

  $('input[type=file]').each(function(i){
    var filedata = new FormData();
    var files = $(this)[0].files;

    $.each(files, function(key, value) {
      filedata.append(key, value);
    });

    $.ajax({
      url: '/apps/uploadFile.groovy',
      type: 'POST',
      data: filedata,
      cache: false,
      processData: false,
      contentType: false,
    }).done(function(data){

      if(data.search("EMPTY") == -1){
        containsfiles = true;
        ajaxdata["FILE_"+fileAmountCounter] = data;
        fileAmountCounter++;
      }

      loopcounterforallfileinputswithfile++;
      if(loopcounterforallfileinputswithfile == total){
        formSend();
      }
    });
  });
}

```

Kuva 7: Tiedoston lähettäminen käyttöliittymässä.

4.1.2 Käyttäjän rekisteröityminen

En halunnut luoda palveluun omaa rekisteröitymistä käyttäjille, koska se nostaa käyttäjien kynnystä liittyä palveluun ja mahdollistaa tarjouspyynnöittömien käyttäjien palvelussa olemisen, minkä takia rekisteröinti luotiin tarjouspyynnön lähettämisen yhteyteen. Tarjouspyynnön lähetyksen jälkeen palvelu kysyy asiakkaalta sähköpostiosoitetta. Sähköpostiosoitetta etsitään palvelusta, mutta jos sitä ei löydy, pyydetään käyttäjää menemään sähköpostiinsa ja aktivoimaan käyttäjätunnus linkin kautta (Kuva 8). Kun asiakas painaa sähköpostissa olevaa linkkiä, pyydetään häneltä vielä salasana, jonka jälkeen tarjouspyyntö muutetaan julkiseksi ja aktivoidaan käyttäjätunnus.

Tarvitsemme vielä sähköpostiosoitteesi tilisi aktivointia varten. Tai jos olet jo aiemmin rekisteröitynyt palveluumme, käyttäjän tunnistautumista varten. Kiitos!

Sähköpostiosoitteesi:

Ilmoita remontti

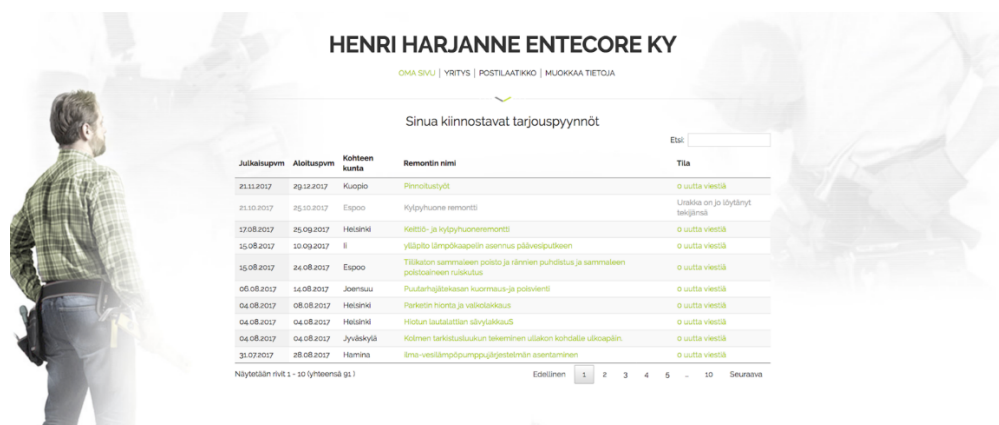
Kuva 8: Sähköpostin pyytäminen tarjouspyynnön aktivointia tai tunnistautumista varten

Rekisteröityneitä käyttäjiä pyydetään antamaan salasana, että voidaan varmistaa käyttäjätunnuksen omistajuus ennen kuin tarjouspyyntö muutetaan julkiseksi. Jo kirjautuneet käyttäjät voivat luoda tarjouspyyntöjä ilman salasanan tai sähköpostiosoitteen kirjoittamista.

4.2 Omat sivut

Palveluun tulevien tarjouspyyntöjen tilan selkeyttämiseksi luotiin käyttäjille omat sivut. Omien sivujen tarkoitus on toimia mahdollisimman samalla periaatteella niin tarjouspyynnön lähettäjän kuin siihen vastaajan puolella. Yhdellä silmäyksellä täytyy saada selville, onko tarjouspyynnön tai tarjouksen tila muuttunut.

Urakoitsijan tärkeimmiksi toiminnoiksi omilla sivuilla valikoitui uusien viestien esille tuleminen ja mahdollisuus nähdä yhdellä hiiren painalluksella uusin viesti. Tarjouspyynnöllä voi olla viisi eri tilaa; olet saanut uuden viestin, tarjous lähetetty, tarjous hyväksytty tai jonkin toisen urakoitsijan tarjous on hyväksytty. Eri tiloilla olevan tarjouspyynnön klikkaaminen ohjaa eri paikkaan. Esimerkiksi, jos olet saanut uuden viestin, ohjataan suoraan keskusteluun (Kuva 9).



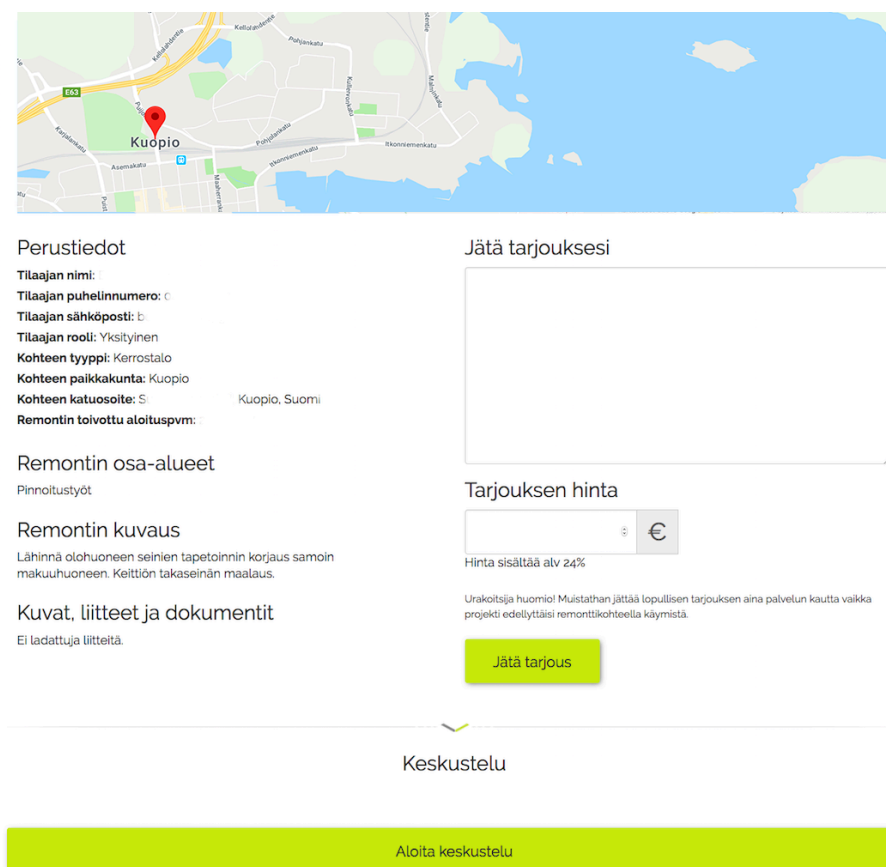
Kuva 9: Palvelun omat sivut yritykselle.

Tarjouspyynnön luojalle eli kuluttajalle tärkeimmät toiminnot omilla sivuilla ovat saada tietoja urakoitsijoiden lähettämistä viesteistä, joista voidaan ohjata suoraan keskusteluun ja nähdä helposti saapuneet tarjoukset.

4.3 Viestikeskus ja tarjouksen jättäminen

Viestikeskus suunniteltiin tukemaan tarjouksen lähettämistä ja mahdollistamaan remontin lisätietojen kysymisen palvelun kautta, joten vain urakoitsija voi aloittaa keskustelun (Kuva 10). Tämä myös säästää urakoitsijoita ylimääräisiltä viesteiltä. Viestikeskus on

reaaliaikainen ja siellä on mahdollista lähettää pdf- ja kuvatiedostoja. Keskustelu sijaitsee tarjouksen luomissivulla, joten viestiä kirjoittaessa voidaan tarkastella kohteen tietoja ja heti vastauksen saapuessa voidaan luoda tarjous viestien perusteella.



Perustiedot

Tilaajan nimi:

Tilaajan puhelinnumero:

Tilaajan sähköposti:

Tilaajan rooli: Yksityinen

Kohteen tyyppi: Kerrostalo

Kohteen paikkakunta: Kuopio

Kohteen katuosoite: Kuopio, Suomi

Remontin toivottu aloituspvm:

Remontin osa-alueet

Pinnostustyöt

Remontin kuvaus

Lähinnä olohuoneen seinien tapetoinnin korjaus samoin makuuhuoneen. Kellion takaseinän maalaus.

Kuvat, liitteet ja dokumentit

Ei ladattuja liitteitä.

Jätä tarjouksesi

Tarjouksen hinta

€

Hinta sisältää alv 24%

Urakoitsija huomio! Muistathan jättää lopullisen tarjouksen aina palvelun kautta vaikka projekti edellyttäisi remonttikohdeella käymistä.

Jätä tarjous

Keskustelu

Aloita keskustelu

Kuva 10: Tarjouspyynnön katsominen sekä viestin ja tarjouksen lähettäminen. Tarjouspyynnön tiedot poistettu.

Ohjelmakoodiin viestien lähetykseen luotiin tarkastelija, joka tarkastelee viestin tilaa ja kuka viestin on nähnyt, jotta palvelun omille sivuille saadaan helposti luotua ilmoitus uusista viesteistä ja ilmoitus poistuu, kun se on luettu. Viestin lähetyksessä tarkastellaan aina viestin lähettäjää ja asetetaan toiselle osapuolelle viestin nähnyt -tilaksi epätosi (Kuva 11). Kun viestit ladataan luettavaksi, tarkastellaan samalla pyynnöllä, onko viestejä, joita käyttäjä ei ole vielä nähnyt ja merkitään niiden tilaksi nähnyt (Kuva 12).


```

message.put("CONVERSATIONID", new ObjectId(args[0].get("convaid")));
message.put("TEXT", args[0].get("text"));//+ logiikka jolla estetään sivustolle mahdollisesti tulevat scripti hyökkäykset);
message.put("USERID", args[0].get("userid")?:null);
message.put("CONTRID", args[0].get("contrid")?:null);
if (args[0].get("uploadedfile")){
    message.put("UPLOADEDFILE", args[0].get("uploadedfile")?:null);
}
if (args[0].get("userid")){
    message.put("USERSEEN", true);
    message.put("CONTRSEEN", false);
}
if (args[0].get("contrid")){
    message.put("CONTRSEEN", true);
    message.put("USERSEEN", false);
}
message.save("message");

```

Kuva 11: Tarkastellaan viestin lähetettäessä, kumpi osapuoli on kyseessä viestien lähetyksessä ja asetetaan vastakkaisen osapuolen muuttujan tila epätodeksi.

```

if (Session.current().get("role").equals("user")){
    if (!message.get("USERSEEN")){
        message.put("USERSEEN", true);
        message.save("message");
    }
    if (message.get("USERID")){
        output += "" <div class="you"><span class="details">Sinä - ${message.get("created").f
    }else{
        output += "" <div class="notyou"> <span class="details">${contr.get("COMPANY_NAME")}
    }
}

```

Kuva 12: Tarkastellaan kumpi osapuoli on kyseessä ja merkitään sen mukaan muuttujan tila nähdynksi.

Tarjouksen luomisesta haluttiin yksinkertainen. Urakoitsija antaa vain lisätiedot, joihin hinta perustuu ja itse hinnan (Kuva 10). Tarjouksen jättämisestä lähetetään aina ilmoitus sähköpostiin ja palvelun omille sivuille, jossa kuluttaja voi tarkastella tarjouksen tietoja. Kun viestikeskustelu on avattu urakoitsijan toimesta, voi urakoitsijalta pyytää vielä lisätietoja.

4.4 Laskutus ja raportointi

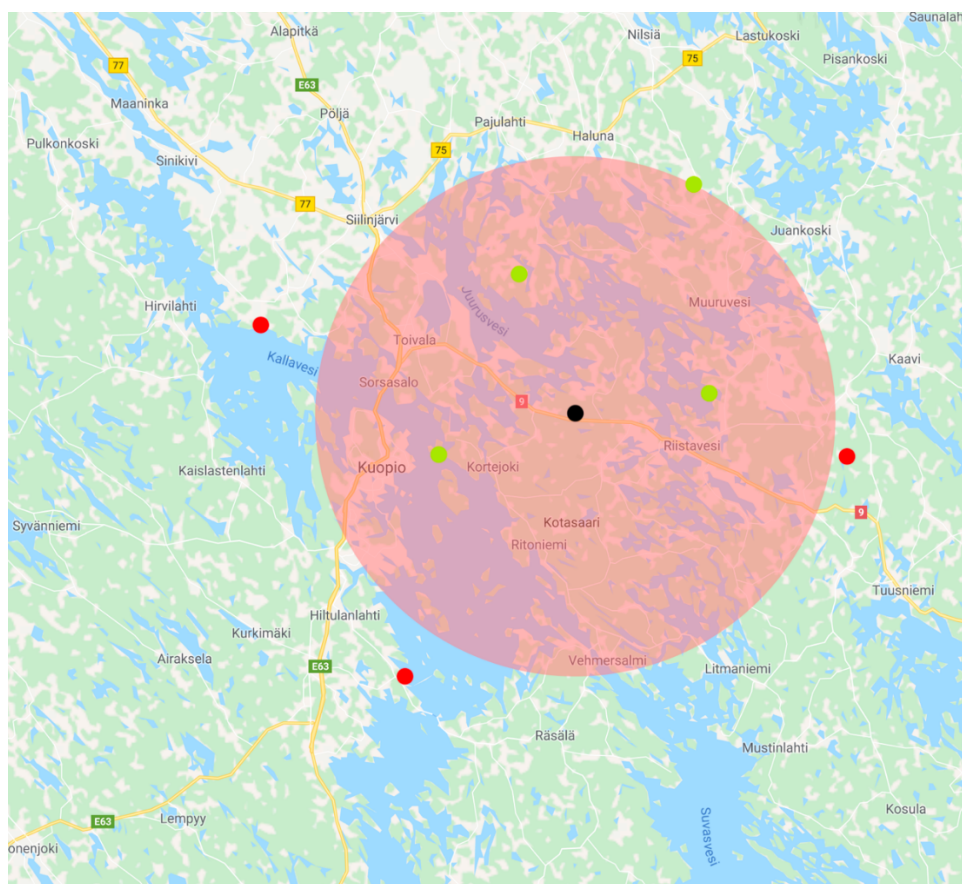
Helpottamaan palvelussa kulkevia laskuja suunniteltiin ja toteutettiin automaattinen laskutus. Lasku lähetetään urakoitsijalle heti toteutuneesta sopimuksesta. Lasku luodaan valmiin pdf-pohjan päälle ja lähetetään se sähköpostiin liitteenä. Myös remonttiapua hakevalle osapuolelle lähetetään sähköposti palvelun käytöstä ja muistutetaan urakoitsijan arvioimisesta. Laskulle lasketaan ohjelmallisesti summa ja eräpäivä, minkä takia tämä osa vaati erityisen paljon testaamista, jotta vääriltä laskuilta vältytään.

Palvelun kasvua ja toimintaa helpottavaksi työkaluksi luotiin raporttien generointiin mahdollisuus Horizon Interactive Oy:lle. Horizon Interactive Oy:n kanssa yhteistyössä mietityt raportit helpottavat seuraamaan muun muassa toteutuneiden remonttien, lähetettyjen laskujen ja luotujen tarjousten määrää. Datan ollessa Excel-muodossa on mahdollista datasta muodostaa erilaisia visualisaatioita data-analytiikka -ohjelmistoilla ketterästi ilman tietokantaintegraatioita.

4.5 Kiinnostavat tarjouspyynnöt

Palvelun oleellisena osana toimii kohdennetut tarjouspyynnöt. Tarjouspyynnöt näytetään vain siitä maantieteellisesti mahdollisesti kiinnostuneille urakoitsijoille, joten suunnittelin palveluun algoritmin, joka valitsee kiinnostavat tarjouspyynnöt urakoitsijalle automaattisesti. Algoritmin suurimpana erottelijana toimii maantieteellinen sijanti ja alue, jolla urakoitsija on valmis tekemään työtä. Kuvassa 13 pisteet ovat algoritmin valitsemia urakoitsijaa kiinnostavia tarjouspyyntöjä ennen maantieteellistä suodatusta ja keskellä oleva punainen ympyrä havainnollistaa urakoitsijan valitsemaa työskentelyaluetta. Musta piste punaisen alueen keskellä on urakoitsija, jonka ympyrällä olevat vihreät pisteet ovat algoritmin valitsemia tarjouspyyntöjä maantieteellisen suodatuksen jälkeen. Nämä tarjouspyynnöt urakoitsija näkee palvelun omilla sivuilla ja saa niiden palveluun lähettämistä ilmoituksen.

Algoritmia varten tutkin erilaisia jo olemassa olevia seuranhakupalveluita ja niiden logiikkaa yhdistää ihmisiä kiinnostuksen kohteiden ja sijainnin perusteella. Teknisenä apuna käytin MongoDB:n valmiita työkaluja maantieteelliseen hakuun. Muita algoritmin käyttämiä parametreja en opinnäytetyössä voi avata asiakkaan pyynnöstä.



Kuva 13: Algoritmin toiminta selkeytettynä.

5 YHTEENVETO

Työn tavoitteena oli luoda Horizon Interactive Oy:lle Remonttimarkkinat-välityspalvelu, jonka ympärille voidaan rakentaa liiketoiminta. Vaatimusmäärittelyn lopputulos ja lopullinen tuote erosivat hieman toisistaan. Asiakkaan tarve tarkentui kehityksen aikana, kuten ketterän kehityksen iteraatiokierroksien kautta on tarkoituskin, ja palvelun muovautui vastaamaan asiakkaan tarpeita.

Oma roolini projektissa oli vastata teknisestä kehityksestä, suunnittelusta ja aikataulussa pysymisestä. Opin projektin aikana paljon asiakkaan kanssa työskentelystä ja asiakkaan vaatimusten kasvavamisesta projektin edetessä. Projektin aikana palvelun toiminnot venyivät hieman alkuperäisen suunnitelman ulkopuolelle, mutta sain kuitenkin ohjattua asiakasta ja projektia oikeaan suuntaan alkuperäisen aikataulun sisällä. Jälkikäteen ajateltuna olisin tuonut asiakkaalle selkeämmin esille kehitysmenetelmän ja sopinut projektipalaverit tietyin aikavälein, jotta projekti olisi pysynyt vielä paremmin kasassa.

Palvelu saatiin valmiiksi asiakkaan kanssa yhteistyössä suunnitellun aikataulun sisällä. Jälkikäteen palvelua ei ole jouduttu korjailemaan, koska Horizon Interactive Oy oli todella suuresti mukana jo kehitysvaiheiden testauksessa. Palvelulle ollaan suunnittelemassa mahdollista monistusta eri aloille pienten muutosten saattelemana.

LÄHTEET JA TUOTETUT AINEISTOT

Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D. (2001) *Manifesto for Agile Software Development* <http://agilemanifesto.org/>

Bootstrap, [Viitattu 19.3.2018] Saatavissa: <https://getbootstrap.com/docs/4.0/layout/grid/>

Build With, *jQuery Usage Statistics* [Viitattu 19.3.2018] Saatavissa: <https://trends.builtwith.com/javascript/jquery>

Groovy, [Viitattu 22.3.2018] Saatavissa: <http://groovy-lang.org/>

Győrödi, C., Győrödi, R., Pecherle, G., Olahhttp, A. (2015) *A comparative study: MongoDB vs. MySQL*, Engineering of Modern Electric Systems (EMES), 2015 13th International Conference, DOI: 10.1109/EMES.2015.7158433

jQuery, *How jQuery works* [Viitattu 2.4.2018] Saatavissa: <https://learn.jquery.com/about-jquery/how-jquery-works/>

MongoDB, [Viitattu 20.3.2018] Saatavissa: <https://www.mongodb.com/scale/relational-vs-non-relational-database>

Pham, A., Pham, P. (2012) *Scrum in Action: Agile Software Project Management and Development*, Course Technology PTR, Boston USA

Play, [Viitattu 22.3.2018] Saatavissa: <https://www.playframework.com/documentation/2.6.x/Philosophy>

Tilastokeskus (2018) *Tilasto: Rakentamisen liikevaihtokuvaaja, Maa- ja vesirakentamisen liikevaihdon kasvu hiipui vuonna 2017 – muut rakentamisen toimialat jatkoivat kasvuaan* ISSN=1798-5919. Helsinki: Tilastokeskus [viitattu: 8.4.2018] Saatavissa: http://www.stat.fi/til/rlv/2017/12/rlv_2017_12_2018-03-13_kat_001_fi.html

TIOBE, *TIOBE Index for April 2018* [Viitattu 2.4.2018] Saatavissa: <https://www.tiobe.com/tiobe-index/>

Tom, E. (2016) *10 Rules of Good UI Design to Follow On Every Web Design Project*, [Viitattu 20.3.2018] Saatavissa: <https://www.elegantthemes.com/blog/resources/10-rules-of-good-ui-design-to-follow-on-every-web-design-project>

Urakkamaailma (2018) [Viitattu 9.4.2018] Saatavissa: <https://www.urakkamaailma.fi/>